Docket No.: 1015-011 ***PATENT***
Express Mail Label No.: EV483417455US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Fred Albert Dykins et al.	: Confirmation No.:	1215
Serial No.:	09/484,865	: Art Unit:	2127
Filed:	1/18/2000	: Examiner:	Kenneth Tang
For:	PROGRAMMER/FEEDER SYSTEM TASK LINKING PROGRAM	:	

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

The following Appeal Brief is submitted pursuant to the Notice of Appeal filed May 6, 2005 in the above-identified Application.

(1) ***Real party in interest:***

The real party in interest is Data I/O Corporation, having its principal place of business in Redmond, Washington.

(2) ***Related appeals and interferences:***

There are no known related appeal or interference cases.

(3) ***Status of claims:***

Claims 1-4, 7-16, and 19-50, the only claims pending, stand under final rejection, from which rejection this Appeal is taken.

Claims 1-4, 7-16, and 19-25 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, second paragraph, set forth in the final rejection.

Claims 30-32, 34-36, 42-44, and 46-48 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

(4) ***Status of amendments:***

No amendments have been filed subsequent to the final rejection of March 8, 2005.

(5) ***Explanation of the invention:***

Element numbers are not required since the invention relates to a method and MPEP 601.01(f) states in relevant part:

> "It has been USPTO practice to treat an application that contains at least one process or method claim as an application for which a drawing is not necessary for an understanding of the invention under 35 U.S.C. 113 (first sentence). ..." [underlining for clarity]

Without being limiting, the following indicates within the square brackets where an exemplary explanation is given in the Specification for each of the method steps.

1. A method for processing microdevices comprising:

providing a computer system having processing information related to the microdevices as a task [page 4, lines 24-25];

providing a legacy processing system [page 4, lines 25-26];

providing a non-legacy processing system for operating independently from the computer system [page 4, lines 26-28];

providing the task from the computer system to the legacy processing system with constant interaction therebetween [page 6, lines 28-31, page 8, lines 27-29];

providing the task from the computer system to the non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system [page 10, lines 19-21];

developing return non-legacy information resulting from the non-legacy processing system using the task [page 10, lines 21-23];

returning the return non-legacy information to the computer system [page 10, lines 24-26];

providing processing system setup and shutdown parameters [page 7, lines 12-15; pasge 8, lines 18-22];

providing processing system process-specific parameters [page 7, lines 24-25];

providing the processing system setup parameters to the legacy processing system and the non-legacy processing system [page 7, line 16; page 9, line 29-31];

providing the processing system shutdown parameters to the non-legacy processing system simultaneously with the processing system setup parameters [page 10, lines 7-8];

providing the number of processed microdevices to be output from the legacy processing system and the non-legacy processing system [page 7, lines 21-23];

providing processing system process-specific parameters to the legacy processing system and the non-legacy processing system [page 7, lines 24-25];

controlling the handling of the microdevices [page 7, line 31, through page 8, line 2];

processing the microdevices [page 8, lines 3-5];

providing the processing system shutdown parameters to the legacy processing system [page 8, lines 21-22];

providing a number of microdevices [page 8, lines 18-19];

determining the number of microdevices processed [page 8, lines 23-26; page 10, lines 15-18];

determining the number of microdevices handled [page 8, lines 23-26; page 10, lines 15-18]; and

developing statistics from the number of microdevices processed and handled [page 8, lines 23-26; page 10, lines 15-18].

13. A method for processing and programming programmable microdevices comprising:

providing a computer system having processing information and programming information related to the programmable microdevices combined as a task in the computer system [page 4, lines 24-25; page 5,lines 25-30];

providing a legacy processing system [page 4, lines 25-26];

providing a programmer/feeder system for operating independently from the computer system [page 4, lines 26-28];

providing the task from the computer system to the programmer/feeder system [page 6, lines 28-31];

performing the task by the programmer/feeder system independent of the computer system by processing and programming the programmable microdevices [page 10, lines 19-21];

developing return programmer/feeder information resulting from the programmer/feeder system using the processing information [page 10, lines 21-23];

returning the return programmer/feeder information to the computer system [page 10, lines 24-26];

providing processing system setup and shutdown parameters [page 7, lines 12-15; pasge 8, lines 18-22];

providing processing system process-specific parameters [page 7, lines 24-25];

providing the processing system setup parameters to the legacy processing system and the programmer/feeder system [page 7, line 16; page 9, line 29-31];

providing the processing system shutdown parameters to the programmer/feeder system simultaneously with the processing system parameters [page 10, lines 7-8];

providing the number of processed programmable microdevices to be output from legacy processing system and the programmer/feeder system [page 7, lines 21-23];

providing the processing system process-specific parameters to legacy processing system and the programmer/feeder system [page 7, lines 24-25];

controlling the handling of the programmable microdevices [page 7, line 31, through page 8, line 2];

programming the programmable microdevices [page 8, lines 3-5];

providing the processing system shutdown parameters to the legacy processing system [page 8, lines 21-22];

providing a number of programmable microdevices page 8, lines 18-19];

determining the number of programmable microdevices processed [page 8, lines 23-26; page 10, lines 15-18];

determining the number of programmable microdevices handled [page 8, lines 23-26; page 10, lines 15-18]; and

developing statistics from the number of programmable microdevices processed and handled [page 8, lines 23-26; page 10, lines 15-18].

26. A method for processing microdevices comprising:

providing a computer system having processing information related to the microdevices as a task [page 4, lines 24-25];

providing a legacy processing system [page 4, lines 25-26];

providing a non-legacy processing system [page 4, lines 26-28];

providing the task from the computer system to the legacy processing system with constant interaction therebetween [page 6, lines 28-31, page 8, lines 27-29];

providing the task from the computer system to the non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system [page 10, lines 19-21];

developing return non-legacy information resulting from the non-legacy processing system using the task [page 10, lines 21-23]; and

returning the return non-legacy information to the computer system [page 10, lines 24-26].

27. The method as claimed in claim 26 additionally comprising:

providing a microdevice programming system in the legacy processing system, the legacy processing system having an on-line connection with said computer system [page 4, lines 26-28]; and

programming the microdevices in the microdevice programming system using the task provided through the on-line connection from the computer system to the processing system [page 8, lines 3-5].

28. The method as claimed in claim 26 additionally comprising:

providing an operator mode [page 6, lines 10-11];

providing a microdevice programming system in the non-legacy processing system, the microdevice programming system standing alone from the computer system [page 4, lines 26-28];

using the processing information for the task in the operator mode in the non-legacy processing system independent from the computer system [page 6, lines 22-24];

returning return information in the operator mode from the non-legacy processing system using portable medium to the computer system [page 10, lines 21-23]; and

storing the return information in the computer system [page 10, lines 15-18].

29. The method as claimed in claim 26 additionally comprising:

providing an administrator mode [page 5, lines 3-5];

providing programming information related to the task in the administrator mode [page 5, lines 8-12];

editing the processing and programming information related to the task in the administrator mode [page 5, lines 13-16]; and

storing the processing and programming information related to the microdevices for the legacy processing system and the non-legacy processing system as the task in the administrator mode [page 5, line 24].

33. The method as claimed in claim 26 additionally comprising:

combining a plurality of tasks to define a kit [page 5, lines 29-30]; and

performing the processing of a kit through the off-line connection [page 10, lines 19-21].

37. The method as claimed in claim 26 additionally comprising:

providing an administrator mode [page 5, lines 13-16]; and

protecting provision of the operator mode using a password input in the administrator mode [page 5, lines 17-23].

38. A method for processing and programming programmable microdevices comprising:

providing a computer system having processing information and programming information related to the programmable microdevices combined as a task in the computer system [page 4, lines 24-25];

providing a legacy processing system [page 4, lines 25-26];

providing a programmer/feeder system [page 4, lines 26-28];

providing the task from the computer system to the programmer/feeder system [page 6, lines 28-31, page 8, lines 27-29];

performing the task by the programmer/feeder system independent of the computer system by processing and programming the programmable microdevices [page 10, lines 19-21];

developing return programmer/feeder information resulting from the programmer/feeder system using the processing information [page 10, lines 21-23]; and

returning the return programmer/feeder information to the computer system [page 10, lines 24-26].

39. The method as claimed in claim 38 additionally comprising:

providing a microdevice programming system in the programmer/feeder system, the programmer/feeder system having an on-line connection with said computer system [page 4, lines 26-28]; and

performing the task by the programmer/feeder dependent on the computer system using programming information obtained through the on-line connection [page 8, lines 3-5].

40. The method as claimed in claim 38 additionally comprising:

providing an operator mode [page 6, lines 10-11];

using portable memory medium to provide the task in the operator mode to the programmer/feeder system independent from the computer system [page 9, lines 16-22];

returning return programmer/feeder information in the operator mode using the portable memory medium to the computer system [page 10, lines 21-23]; and

storing the return programmer/feeder information in the computer system [page 10, lines 15-18].

41. The method as claimed in claim 38 additionally comprising:

providing an administrator mode [page 5, lines 3-5];

providing the processing and programming information related to the task in the administrator mode [page 5, lines 8-12];

editing the processing and programming information related to the task in the administrator mode [page 5, lines 13-16]; and

storing the processing and programming information related to the programmable microdevices for the legacy processing system and the programmer/feeder system in the administrator mode [page 5, line 24].

45. The method as claimed in claim 38 additionally comprising:

combining a plurality of tasks to define a kit [page 5, lines 29-30]; and

performing the programming of a kit in the legacy processing system and the programmer/feeder [page 10, lines 19-21].

49. The method as claimed in claim 38 additionally comprising:

providing an administrator mode [page 5, lines 13-16]; and

protecting provision of the operator mode using a password input in the administrator mode [page 5, lines 17-23].

50. The method as claimed in claim 38 additionally comprising:

providing information for affecting changes selected from a group consisting of software, firmware, and a combination thereof by using the portable memory medium [page 9, lines 16-22].

(6) Grounds for Rejections:

Rejection #1:

Claims 1-4, 7-16, and 19-50 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Rejection #2:

Claims 26-29 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tyner et al. (US 6,272,618 B1, hereinafter Tyner) in view of Bodnar et al. (US 6,658,268 B1, hereinafter Bodnar).

Rejection #3:

Claims 38-41 and 49-50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tyner et al. (US 6,272,618 B1, hereinafter Tyner) in view of Bodnar et al. (US 6,658,268

B1, hereinafter Bodnar), and further in view of Coburn et al. (US 2002/0120921 Al, hereinafter Coburn).

Rejection #4:

Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tyner et al. (US 6,272,618 B1, hereinafter Tyner) in view of Bodnar et al. (US 6,658,268 B1, hereinafter Bodnar), and further in view of Kenik et al. (US 4,821,197, hereinafter Kenik).

Rejection #5:

Claim 45 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tyner et al. (US 6,272,618 B1, hereinafter Tyner) in view of Bodnar et al. (US 6,658,268 B1, hereinafter Bodnar), further in view of Coburn et al. (US 2002/0120921 Al, hereinafter Coburn), and further in view of Kenik et al. (US 4,821,197, hereinafter Kenik).

(7) ***Arguments:***

Rejection #1:

Arguments:

It is respectfully submitted that claims 1-4, 7-16, and 19-50 are improperly rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding claim 1, the Examiner states in the Final Office Action of March 8, 2005 (hereinafter the "Final Rejection"):

> "In claim 1, "processing microdevices" (line 26) is indefinite because it is not made explicitly clear in the claim language if the computer, legacy, and/or the non-legacy system processes the microdevices."

Appellants respectfully disagree. It has long been settled that:

> "[B]y statute, 35 U.S.C. 112, Congress has placed no limitations on *how* an applicant claims *his invention*, so long as the specification concludes with claims which particularly point out and distinctly claim that invention." *In re* Pilkington, 411 F.2d 1345, 1349-50, 162 USPQ 145, 148 (C.C.P.A. 1969) (quoting *In re* Steppan, 394 F.2d 1013, 1019, 156 USPQ 143, 148 (C.C.P.A. 1967) (emphasis in original). "Moreover, it is not the normal function of a claim to disclose the invention, but to point out the features of novelty in the invention as disclosed in the specification and drawing of the application." [underlining for clarity] Bocciarelli v. Huffman, 232 F.2d 647, 109 USPQ 385, 388 (C.C.P.A. 1956).

Specification page 6, lines 22-24, describes how the processing of microdevices occurs for legacy and non-legacy systems:

> "The TaskLink program then proceeds to a "Program Microdevices" block 52. This block will be described in greater detail with reference to FIGs. 3A-3C for on-line systems and FIGs. 4A-4B for off-line systems."

Specification page 6, lines 28-31, discloses the processing by the legacy system and not by the computer:

> "Referring now to FIGs. 3A-3C, therein is shown the legacy and related system processing in the "Program Microdevices" block 52 in FIG. 2. Generally in these systems, the computer on which the TaskLink program resides is directly connected to the programming system by a local area network, such as Ethernet."

Specification page 8, line 30, through page 9, line 6, discloses the processing by the non-legacy system and not the computer:

> "Referring now to FIGs. 4A-4B, therein is shown the programming for a programming system, such as a programmer/feeder system, in the "Process Microdevices" block 52 in FIG. 2. Generally in these systems, the computer system on which the TaskLink program resides is not directly connected to control the programmer/feeder system and is connected to provide and receive information periodically to and from the programmer/feeder system. However, it would be evident to those skilled in the art that nothing herein precludes a direct connection between the computer and the programmer/feeder system by a local area network, but the operation would be one in which there would not constant interaction and contact. This type of contact where the computer is isolated from the programming system while the processing occurs is referred to as an "off-line" connection."

Further, it would be obvious to those having ordinary skill in the art that the legacy and non-legacy processing systems process the microdevices since Appendix 1, claim 1, lines 21-22, and claim 13, lines 21-22, as exemplified by claim 1, state:

> "providing the number of processed microdevices to be output from the legacy processing system and the non-legacy processing system;" [underlining for clarity]

The Examiner continues:

> "It is unclear in (*sic*) the microdevices are a part of the legacy system, the non-legacy system, or neither. There is no link or relationship that connects microdevices to these systems. Claims 13, 26, and 38 are rejected for the same reasons."

Appellants respectfully disagree. It would be obvious to one having ordinary skill in the art from Specification page 10, line 27, through page 11, line 5, that the microdevices are acted on by the legacy and non-legacy systems and are not part of the invention:

> "From the above it will be understood that the present invention is applicable to what can be described as "microdevices". Microdevices include a broad range of electronic and mechanical devices. The best mode describes processing which is programming for programmable devices, which include but are not limited to devices such as Flash memories (Flash), electrically erasable programmable read only memories (E^2PROM), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), and microcontrollers. However, the present invention encompasses processing for all electronic, mechanical, hybrid, and other devices which require testing, measurement of device characteristics, calibration, and other processing operations. For example, these types of microdevices would include but not be limited to devices such as microprocessors, integrated circuits (ICs), application specific integrated circuits (ASICs), micro mechanical machines, micro-electro-mechanical (MEMs) devices, micro modules, and fluidic systems."

Further, the preamble indicates that the invention is a method for processing microdevices; i.e., the microdevices are acted upon by the system and are not elements of the invention.

For the reasons given above, claims 13, 26, and 38 are submitted to be allowable.

Regarding claims 26 and 38 further, it would be obvious to one having ordinary skill in the art that the task instructs the legacy and non-legacy (programmer feeder) systems as the Specification describes in page 3, lines 24-26:

> "The present invention provides a task linking program for using a computer for interacting with on-line and off-line processing systems to perform tasks related to processing microdevices."

Regarding claim 1 still further, the Examiner states in the "Final Rejection":

> "In claim 1, there is no link or relationship established between "non-legacy information" (line 14) and "system setup and shutdown parameters" (line 15) and "system process-specific parameters" (line 16), and therefore, it is not made explicitly clear in the claim language whether or not "non-legacy information" constitutes "system setup and shutdown parameters" and "system process-specific parameters". Claim 13 is rejected for the same reasons."

Appellants respectfully disagree. As stated by Bocciarelli v. Huffman, *supra*: "it is not the normal function of a claim to disclose the invention, but to point out the features of novelty in the invention as disclosed in the specification and drawing of the application." The claimed novelty is in the developing of return non-legacy information and returning it to the computer system as indicated by Appendix 1, claim 1, lines 12-14:

> "developing return non-legacy information resulting from the non-legacy processing system using the task;
> returning the return non-legacy information to the computer system;"
> [underlining for clarity]

The above is supported in Specification page 10, lines 13-14:

> "Upon completion of programming in the "Programming" block 172, the job statistics are returned at "Return" block 174 by the memory card."

Regarding claim 1 additionally, the Examiner states in the "Final Rejection":

> "In claim 1, "providing the number of processed microdevices to be output from the legacy processing system and the non-legacy processing system" (line 21) and "providing processing system process-specific parameters to the legacy processing system and the non-legacy processing system" (line 23) is indefinite because it is not made explicitly clear in the claim language whether the microdevices or computer system provides this. Claim 13 is rejected for the same reasons."

Appellants respectfully disagree. It would be obvious to one having ordinary skill in the art even from the phrases quoted by the Examiner that the overall system, without being

limiting, comprises three basic systems: the legacy processing system, the non-legacy processing system, and the computer system. Since the number and the parameters are provided to two of the three basic systems, by process of elimination if no other, the third basic system, the computer system, must provide the number and parameters. As would be obvious to one having ordinary skill in the art, the microdevices first appear in the preamble of claim 1 to indicate that they are not elements of the claimed invention but something acted upon by the claimed invention.

In any event, this is explained in Specification page 4, lines 23-28:

> "The present invention relates to an overall system (not shown) which consists of three basic systems. First, is the computer system in which the program of the present invention operates. Second, is the legacy processing system, such as a programming system, which is on-line with the computer. And third, is the new processing system, such as a programmer/feeder system, which is generally off-line or generally operates independently from the computer system."

Regarding claims 13, 26, 30, 34, 38, 42, and 46, the Examiner states in the "Final Rejection":

> "Claims 13, 26, 30, 34, 38, 42, and 46 had the same deficiency as claim 1 above. Corrections to claim 1 are also required to overcome the rejection for these claims."

Appellants respectfully submit that the arguments for claim 1 are applicable to claims 13, 26, 30, 34, 38, 42, and 46.

Applicants also respectfully submit that the arguments for claim 1 are applicable to claims 2-4, 7-12, 14-16, 19-25, 27-29, 31-33, 35-37, 39-41, 43-45, and 47-50.

Based on all of the above, it is respectfully submitted that claims 1-4, 7-16 and 19-50 are allowable under 35 U.S.C. §112, second paragraph, based on 35 U.S.C. §112, second paragraph, *In re* Pilkington, *supra,* and Bocciarelli v. Huffman, *supra.*

Rejection #2:

Summary of Tyner:

Tyner teaches a system for handling system management interrupts in a multi-processor computer. When the computer enters system management mode, the method uses the registers of each processor to get currently executing opcode to determine what each processor was doing before the interrupt. The contents of the registers can be used to determine if the current processor caused the system management interrupt. If so, then the method now knows which processor caused the interrupt and can handle the interrupt accordingly. If, however, the processor was not the one that caused the interrupt, or if another processor also caused an interrupt, the method then repeats the above steps for the next processor of the multiprocessor system. Taken as a whole, Tyner teaches how interrupts are handled in a multiprocessor computer, but does not teach or suggest a system for processing microdevices.

Summary of Bodnar:

Bodnar teaches an enhanced "Camel-back" or "Companion" Digital Organizer (CDO) that is designed to interface to a cellular telephone. Taken as a whole, Bodnar teaches a combined cell phone and personal organizer.

Arguments:

It is respectfully submitted that claims 26-29 and 37 are improperly rejected under 35 U.S.C. §103(a) as being unpatentable over Tyner in view of Bodnar.

The Examiner states in the Final Rejection of March 8, 2005, (hereinafter the "Final Rejection) item 5:

> "Tyner teaches a method for processing microdevices comprising:"

Appellants respectfully disagree. Neither Tyner nor Bodnar teaches or suggests the claimed "processing", which is defined in Specification page 10, line 29, through page 11, line 2:

> "The best mode describes processing which is programming for programmable devices, which include but are not limited to devices such as

> Flash memories (Flash), electrically erasable programmable read only memories (E^2PROM), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), and microcontrollers. However, the present invention encompasses processing for all electronic, mechanical, hybrid, and other devices which require testing, measurement of device characteristics, calibration, and other processing operations." [underlining for clarity]

A fundamental principle contained in 35 U.S.C. 112, second paragraph is that applicants are their own lexicographers. They can define in the claims what they regard as their invention essentially in whatever terms they choose so long as the terms are not used in ways that are contrary to accepted meanings in the art. Applicant may use functional language, alternative expressions, negative limitations, or any style of expression or format of claim which makes clear the boundaries of the subject matter for which protection is sought. As noted by the Court in *In re* Swinehart, 439 F.2d 210, 160 USPQ 226 (CCPA 1971), a claim may not be rejected solely because of the type of language used to define the subject matter for which patent protection is sought.

Therefore, none of the Examiner cited sections teaches or suggests the claimed "processing" of microdevices as would be understood by those having ordinary skill in the art from a reading of the Specification. Tyner teaches an interrupt signal handling scheme within a multiprocessor computer system and Bodnar teaches a combination cell phone and personal organizer. Taken as a whole, Tyner and Bodnar would not teach one having ordinary skill in the art the claimed invention for processing a microdevice. As stated by the Court of Appeals for the Federal Circuit:

> "Although the PTO must give claims their broadest reasonable interpretation, this interpretation must be consistent with the one that those skilled in the art would reach." *In re* Cortright, 165 F.3d 1353, 1358 (Fed. Cir. 1999), cited in *In re* American Academy of Science Tech Center, CAFC 03-1531, May 13, 2004."

Thus, Tyner in teaching the handling of interrupts in a computer system does not teach or suggest:

providing a computer system having processing information related to the microdevices as a task in Tyner col. 6, lines 32-34 (does not mention claimed element), col. 3, lines 5-14 (describing a chipset) and 45-55 (describing operation of system processor), Fig. 1, item 16 (item 16 is a chip set);

providing a legacy processing system in Tyner col. 6, lines 32-34 (not a disclosure because citation is in the Tyner claims);

providing a non-legacy processing system in Tyner col. 6, lines 32-34 (not a disclosure because citation is in the Tyner claims);

providing the task from the computer system to the legacy processing system with constant interaction therebetween in Tyner col. 6, lines 32-34 (not a disclosure because citation is in the Tyner claims) , col. 3, lines 34-40 (does not describe a providing a task); or

performing a task by the non-legacy processing system independent of the computer system in Tyner col. 6, lines 32-34 (not a disclosure because citation is in the Tyner claims).

The Examiner states in the Final Rejection item 6:

"Tyner fails to explicitly teach:

providing the task from the computer system to a non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system, developing return non-legacy information resulting from the non-legacy processing system using the task, and returning the return non-legacy information to the computer system."

Appellants respectfully agree. It is further respectfully pointed out that Tyner discloses a legacy keyboard for providing commands to a computer not a task from the computer to the keyboard. This disclosure is in Tyner col. 1, lines 35-47:

> "To resolve these problems, the computer must implement device support outside of the operating environment. For example, if a computer uses a USB port for a USB keyboard while implementing a PS/2 operating environment (a PS/2 operating environment expects a PS/2 type keyboard), the computer's basic input/output system ("BIOS") can intercept data and software access to and from specific ports associated with the keyboard. The BIOS thereby "translates" the commands between the newer USB keyboard and the operating environment for use with older (or "legacy") keyboards by examining the internal processor ports and registers and "routing" data accordingly." [underlining for clarity]

The Examiner continues in the Final Rejection item 7:

> "It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of providing the task from the computer system to the non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system,

> developing return non-legacy information resulting from the non-legacy processing system using the task, and returning the return non-legacy information to the computer system to the existing system and method of Tyner in order to transfer instructions and information, thus making both the computer and non-legacy system "smarter" *(col. 17, lines 33-63).*"

Appellants respectfully disagree because Bodnar does not teach or suggest making a computer and a non-legacy telephone "smarter", but relates to how a telephone docking unit operates as explained in Bodnar col. 17, lines 33-63:

> "Built-in docking units are docking units for non-legacy phones... To the extent that different target phones do not use a same communication protocol, a main unit that is to be docked to multiple docking units, for multiple target phones, may be loaded with...phone-model-specific... software. Upon docking..., such a main unit determines...the model of the host phone...and uses the software in the main unit's memory that is specific to the model of the host phone. In a particular embodiment, the phone-model-specific software is created and sold with each new model of docking unit... In an alternate embodiment, each docking unit may include an amount of memory (e.g., flash-based memory) that contains phone-model-specific software for instructing the main unit to work with the docking unit's compatible model(s) of phone..."

Assuming *arguendo* that all the claimed elements are present, it is respectfully submitted that claim 26 is allowable under 35 USC §103(a) as being unpatentable over Tyner in view of Bodnar because the above does not provide a motivation for the combination and because MPEP §2143.01 states:

> "The combination of references taught every element of the claimed invention, however without a motivation to combine, a rejection based on a *prima facie* case of obvious was held improper. The level of skill in the art cannot be relied upon to provide the suggestion to combine references. *Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 50 USPQ2d 1161 (Fed. Cir. 1999)"

Regarding claim 27, the Examiner states in the Final rejection item 8:

> "Tyner teaches:
>
> providing a microdevice progrnmming (*sic*) system in the legacy processing system, the legacy processing system having an on-line connection with said computer system *(col. 5, line 17, Fig. 1, item 30, col. 6, lines 32-34);* and
>
> programming the microdevices in the microdevice progrnmming (*sic*) system using the task provided through the on-line connection from the computer system to the processing system *(col. 5, line 17, Fig. 1, item 30, col. 6, lines 32-34).*"

Appellants respectfully disagree. Tyner does not teach or suggest the Examiner's statement in col. 5, line 17, which states:

"...standby or wake-on-LAN. This generates an event that..."

Tyner does not teach or suggest the Examiner's statement in FIG. 1, item 30, which is described in Tyner col. 3, line 16, as:

"...such as a network interface card 30,"

Tyner does not teach or suggest the Examiner's statement in Tyner col. 6, lines 32-34, which is not a disclosure because the citation is in the Tyner claims and thus must rely for its disclosure on the Tyner Specification. Further, Tyner col. 6, lines 32-34, merely claims:

> "12. The method of claim 7 wherein the computer system includes a non-legacy component and is running an operating environment that only supports legacy components."

This does not teach or suggest providing a computer system and a legacy and non-legacy processing systems (Tyner has a computer system that includes legacy and non-legacy components); providing a non-legacy processing system; or performing a task independent of the computer system (Tyner' non-legacy component, a keyboard, cannot operate independent of the Tyner computer system).

Regarding claim 28, the Examiner states in the Final rejection item 9:

> "As to claim 28, it is rejected for the same reasons as stated in the rejection of claim 26. In addition, Bodnar teaches a user mode *(see Abstract)*."

Appellants respectfully disagree for the same reasons as claim 26. In addition, a *prima facie* case of obviousness has not been made since there is no motivation given for a combination of a computer handling interrupts to have a user mode since the interrupts need to always be handled using the same method. It is respectfully submitted that each reference has not been taken as whole but only portions of each reference have been combined and this is impermissible because the CAFC has stated:

> "One cannot...pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." *In re* Fritch, 972 F.2d 1260, 23 USPQ2d 1780 (Fed. Cir. 1992).

Regarding claim 29, the Examiner states in the Final rejection item 10:

> "As to claim 29, it is rejected for the same reasons as stated in the rejection of claim 26. In addition, Tyner and Bodnar fail to explicitly teach having an administrator mode. However, "Official Notice" is taken that both the concept and advantages of an administrator is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to an administrator mode to the existing system and method in order to provide control and security."

Appellants respectfully disagree for the same reasons given in claim 26. In addition, a *prima facie* case of obviousness has not been made since there is no motivation given for a combination of a computer handling interrupts to have an administrator mode since the interrupts need to always be handled using the same method in a computer and are not subject to administrator control. It is respectfully submitted that each reference, including "Official Notice", has not been taken as whole but only portions of each reference have been combined and this is impermissible based on *In re* Fritch, *supra*.

Regarding claim 37, the Examiner states in the Final rejection item 11:

> "As to claim 37, Tyner and Bodnar fails (*sic*) to explicitly teach providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode. However, "Official Notice" is taken that both the concept and advantages of providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode to the existing system and method in order to provide security."

Appellants respectfully disagree. A *prima facie* case of obviousness has not been made since there is no motivation given for a combination of a computer handling interrupts to have a administrator mode since the interrupts need to always be handled using the same method. It is respectfully submitted that each reference, including "Official Notice", has not been taken as whole but only portions of each reference have been combined and this is impermissible based on *In re* Fritch, *supra*.

Based on all of the above, it is respectfully submitted that claims 26-29 and 37 are allowable under 35 U.S.C. §103(a) as being patentable over Tyner in view of Bodnar.

Rejection #3:

Summary of Tyner:

Supra.

Summary of Bodnar:

Supra.

Summary of Coburn:

Coburn teaches a simulation method for generating simulation data structures, which can be used by a modeling system to interface between a programmable logic controller (PLC) and a movie module. The method includes importing the simulation information from data constructs and populating data structures. Taken as a whole, Coburn teaches a computerized "development process" of designing, constructing, and debugging a manufacturing process.

Arguments:

Claims 38-41 and 49-50 are improperly rejected under 35 U.S.C. 103(a) as being unpatentable over Tyner in view of Bodnar, and further in view of Coburn.

Regarding claim 38, the Examiner states in item 13 of the Final Rejection:

> "As to claim 38, it is rejected for the same reasons as stated in the rejection of claim 26. In addition, Tyner in view of Bodnar fails to explicitly teach using a programmer/feeder system. However, Coburn teaches using a programmer/feeder system consisting of robots, computers, programmable logic controllers, mills, drills, stamps, clamps, sensors, transfer bars, assemblers, etc. because almost every industry has recognized its advantage that use of automated assembly and machining lines to form and assemble product components and assemblies reduce manufacturing time, reduces product costs, and increases product quality *(page* 1, [0005]) It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of a programmer/feeder system to the existing system of Tyner and Bodnar in order to gain the advantages mentioned above."

Appellants respectfully disagree for the same reasons as given for claim 26. In addition, it is respectfully submitted that Coburn is a computer simulation method (Coburn Title) and does not teach or suggest a programmer/feeder system, which could be combined with Tyner and Bodnar.

Regarding claim 38 further, Applicants respectfully traverse the rejection because the Examiner incorrectly describes Coburn in the Office Action by stating:

> "Coburn teaches using a progammer/feeder (sic) system consisting of robots, computers, programmable logic controllers, mills, drills, stamps, clamps, sensors, transfer bars, assemblers, etc. because almost every industry has recognized its advantage that use of automated assembly and machining lines to form and assemble product components and assemblies reduce manufacturing time, reduces product costs, and increases product quality *(page 1, [0005]).*"

Coburn page 1, para. [0005], is Background and states:

> [0005] A visit to virtually any modern manufacturing facility in the world leaves room for little doubt that assembly and machining lines have become an integral part of the manufacturing process. Robots, computers, programmable logic controllers, mills, drills, stamps, clamps, sensors, transfer bars, assemblers, etc., are more numerous than people in most modern manufacturing facilities. This is because almost every industry has recognized that use of automated assembly and machining lines to form and assemble product components and assemblies reduces manufacturing time, reduces product costs and increases product quality. Hereinafter, automated assembly and machining will be referred to collectively as automated manufacturing."

As would be obvious to those having ordinary skill in the art, a programmer/feeder both programs and feeds microdevices. Coburn para. [0005] describes any modern manufacturing facility but, as shown above, does not teach or suggest the claimed programmer/feeder. Since the limitation is not taught or suggested, Applicants respectfully and timely requested an Examiner Affidavit disclosing the Examiner's personal knowledge regarding this limitation pursuant to 37 CFR §1.104(d)(2), supra, so it may be traversed. Since the Examiner Affidavit has not been provided, it is respectfully submitted that a *prima facie* case of obviousness has not been made.

Regarding claim 39, the Examiner rejected claim 39 for the same reasons as stated in the rejection of claim 27. Appellants respectfully disagree for the same reasons as presented for claim 27, *supra.*

Regarding claim 40, the Examiner rejected claim 40 for the same reasons as stated in the rejection of claim 28. Appellants respectfully disagree for the same reasons as presented for claim 28, *supra.*

Regarding claim 41, the Examiner rejected claim 41 for the same reasons as stated in the rejection of claim 26. Appellants respectfully disagree for the same reasons as presented for claim 26, *supra*.

Further, the Examiner rejected claim 41 by stating:

> "In addition, Tyner, Bodnar, and Coburn fail to explicitly teach having an administrator mode. However, "Official Notice" is taken that both the concept and advantages of an administrator is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to an administrator mode to the existing system and method in order to provide control and security."

Appellants respectfully disagree, because a *prima facie* case of obviousness has not been made since there is no motivation for a combination of a computer handling interrupts to have an administrator mode since the interrupts need to always be handled using the same method and are not subject to change by an administrator. It is respectfully submitted that each reference, including "Official Notice", has not been taken as whole but only portions of each reference have been combined and this is impermissible based on *In re* Fritch, *supra*.

Further, since the limitation is not taught or suggested, Applicants respectfully and timely requested an Examiner Affidavit disclosing the Examiner's personal knowledge regarding this limitation pursuant to 37 CFR §1.104(d)(2), supra, so it may be traversed. Since the Examiner Affidavit has not been provided, it is respectfully submitted that a *prima facie* case of obviousness has not been made.

Regarding claim 49, the Examiner states in the Final Rejection:

> "As to claim 49, Tyner, Bodnar, and Coburn fails to explicitly teach providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode. However, "Official Notice" is taken that both the concept and advantages of providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include providing an administrator mode and protecting provision of the operator mode using a password input in the administrator mode to the existing system and method in order to provide security."

Appellants respectfully disagree, because a *prima facie* case of obviousness has not been made since there is no motivation for a combination of a computer handling interrupts to have an administrator mode since the interrupts need to always be handled using the same

method. It is respectfully submitted that each reference, including "Official Notice", has not been taken as whole but only portions of each reference have been combined and this is impermissible based on *In re* Fritch, *supra*.

Further, since the limitation is not taught or suggested, Applicants respectfully and timely requested an Examiner Affidavit disclosing the Examiner's personal knowledge regarding this limitation pursuant to 37 CFR §1.104(d)(2), supra, so it may be traversed. Since the Examiner Affidavit has not been provided, it is respectfully submitted that a *prima facie* case of obviousness has not been made.

Regarding claim 50, the Examiner states in the Final Rejection item 18:

> "As to claim 50, Bodnar teaches providing information for affecting changes selected from a group consisting of software, firmware, and a combination thereof by using the portable memory medium *(col. 17, lines 33-63)*."

Appellants respectfully disagree. Assuming *arguendo* that all the claimed elements are present in the cited references, it is respectfully submitted that claim 50 is allowable under 35 USC §103(a) as being unpatentable over Tyner in view of Bodnar because the above does not provide a motivation for the combination as required by MPEP §2143.01. As explained previously for Tyner, there is no reason for changing the interrupt handling in a computer once it is established.

Based on all of the above, it is respectfully submitted that claims 38-41 and 49-50 are allowable under 35 U.S.C. 103(a) as being patentable over Tyner in view of Bodnar, and further in view of Coburn.

Rejection #4:

Summary of Tyner:

Supra.

Summary of Bodnar:

Supra.

Summary of Kenik:

Kenik teaches system for manufacturing multiple component assemblies utilizing semi-automatic computer-assisted material handling. The system comprises a plurality of component selection cells each having an alphanumeric display to display a selected assembly model description and a plurality of component bins wherein each bin has an associated numeric display for displaying the quantity of components from that bin required for a selected assembly model. Each numeric display has a means for clearing the display after the required quantity of components has been selected and for generating a control signal in response to a completion of a selected collection of parts for that cell relating to the selected assembly model. A computer controls the alphanumeric and numeric displays responsive to data entered by an operator relating to the selected assembly model number and controls the clearing of the alphanumeric displays in response to the control signal.

Arguments:

Regarding claim 33, the Examiner states in the Final Rejection item 18:

> "As to claim 33, Tyner in view of Bodnar fails to explicitly teach combining a plurality of tasks to define a kit and performing the processing of a kit through the off-line connection. However, Kenik teaches using kits to perform off-line subassemblies *(col. 5, lines 33-44)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the, feature of combining a plurality of tasks to define a kit and performing the processing of a kit through the off-line connection because this allows for tracking, updating and *maintaining inventory (col. 5, lines 33-44)*."

Appellants respectfully disagree. Kenik teaches hardware parts assembled into kits as indicated by the Examiner italicized for "*maintaining inventory*", and does not teach or suggest combining software tasks into a software kit and performing the processing of microdevices using the kit through an off-line connection.

Rejection #5:

Summary of Tyner:

Supra.

Summary of Bodnar:

Supra.

Summary of Coburn:

Supra.

Summary of Kenik:

Supra.

Arguments:

Regarding claim 45, the Examiner states in the Final Rejection item 22:

> "As to claim 45, Tyner, Bodnar, and Coburn fail to explicitly teach combining a plurality of tasks to define a kit and performing the processing of a kit through the off-line connection. However, Kenik teaches using kits to perform off-line subassemblies *(col. 5, lines 33-44).* It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of combining a plurality of tasks to define a kit and performing the processing of a kit through the off-line connection because this allows for tracking, updating and maintaining inventory *(col. 5, lines 33-44).*"

Appellants respectfully disagree. It is respectfully submitted that the claim is allowable for the same reason as for claim 38 over Tyner, Bodnar, and Coburn and as for claim 33 over Kenik. The cited references do not teach or suggest the claimed combinations and the CAFC has held:

> "The question is whether the prior art, considering its scope and content and the level of ordinary skill, must itself suggest the combination of separate elements into the claimed invention in suit, not just whether it illustrates separate elements..." Laitram Corp. v. Cambridge Wire Cloth Co., 226 USPQ 298 at 293n (D. Md. Mag. 1985), aff'd in part, rev'd in part, and remanded, 785 F.2d 292, 228 USPQ 935 (Fed. Cir. 1986), cert. denied, 479 U.S. 820 (1986):

Based on the above, it is respectfully submitted that claim 45 is allowable under 35 U.S.C. 103(a) as being patentable over Tyner in view of Bodnar, further in view of Coburn, and further in view of Kenik.

(8) ***Conclusion and Relief Requested:***

Claims 1-4, 7-16, and 19-50 particularly point out and distinctly claim the subject matter, which the Appellants regard as their invention, and are patentable over the prior art.

Reversal of the Examiner's decision is respectfully requested.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including any extension of time fees, to Deposit Account No. 50-0374 and please credit any excess fees to such deposit account.

Respectfully submitted,

Mikio Ishimaru
Registration No. 27,449

The Law Offices of Mikio Ishimaru
1110 Sunnyvale-Saratoga Rd., Suite A1
Sunnyvale, CA 94087
Telephone: (408) 738-0592
Fax: (408) 738-0881
Date: July 19, 2005

Appendix I – Claims on Appeal (on following page)

CLAIMS ON APPEAL

1. A method for processing microdevices comprising:

providing a computer system having processing information related to the microdevices as a task;

providing a legacy processing system;

providing a non-legacy processing system for operating independently from the computer system;

providing the task from the computer system to the legacy processing system with constant interaction therebetween;

providing the task from the computer system to the non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system;

developing return non-legacy information resulting from the non-legacy processing system using the task;

returning the return non-legacy information to the computer system;

providing processing system setup and shutdown parameters;

providing processing system process-specific parameters;

providing the processing system setup parameters to the legacy processing system and the non-legacy processing system;

providing the processing system shutdown parameters to the non-legacy processing system simultaneously with the processing system setup parameters;

providing the number of processed microdevices to be output from the legacy processing system and the non-legacy processing system;

providing processing system process-specific parameters to the legacy processing system and the non-legacy processing system;

controlling the handling of the microdevices;

processing the microdevices;

providing the processing system shutdown parameters to the legacy processing system;

providing a number of microdevices;

determining the number of microdevices processed;

determining the number of microdevices handled; and

developing statistics from the number of microdevices processed and handled.

2. The method as claimed in claim 1 additionally comprising:

providing a microdevice programming system in the legacy processing system, the legacy processing system having an on-line connection with said computer system; and

programming the microdevices in the microdevice programming system using the task provided through the on-line connection from the computer system to the processing system.

3. The method as claimed in claim 1 additionally comprising:

providing an operator mode;

providing a microdevice programming system in the non-legacy processing system, the microdevice programming system standing alone from the computer system;

using the processing information for the task in the operator mode in the non-legacy processing system independent from the computer system;

returning return information in the operator mode from the non-legacy processing system using portable medium to the computer system; and

storing the return information in the computer system.

4. The method as claimed in claim 1 additionally comprising:

providing an administrator mode;

providing programming information related to the task in the administrator mode;

editing the processing and programming information related to the task in the administrator mode; and

storing the processing and programming information related to the microdevices for the legacy processing system and the non-legacy processing system as the task in the administrator mode.

5. (canceled)

6. (canceled)

7. The method as claimed in claim 1 additionally comprising:

serializing the microdevices; and

maintaining a log of the serialized microdevices.

8. The method as claimed in claim 1 additionally comprising:

combining a plurality of tasks to define a kit; and

performing the processing of a kit in the legacy processing system and the non-legacy processing system.

9. The method as claimed in claim 1 additionally comprising:

providing microdevice information;

providing processing system setup parameters;

providing format information related to the non-legacy processing system;

inputting the number of processed microdevices to be output from the non-legacy processing system;

providing the processing system setup parameters and format to the non-legacy processing system;

transferring the microdevice information from the computer system to the non-legacy processing system;

transferring the processing system format from the computer system to the non-legacy processing system;

processing the microdevices;

obtaining information from the processing of the microdevices; and

transferring the information from the processing of the microdevices to the computer system.

10. The method as claimed in claim 9 wherein the step of:

transferring includes the use of a portable memory medium.

11. The method as claimed in claim 9 wherein the step of:

transferring includes the use of a direct communication connection.

12. The method as claimed in claim 1 including the steps of

providing an administrator mode; and

protecting provision of the operator mode using a password input in the administrator mode.

13. A method for processing and programming programmable microdevices comprising:

providing a computer system having processing information and programming information related to the programmable microdevices combined as a task in the computer system;

providing a legacy processing system;

providing a programmer/feeder system for operating independently from the computer system;

providing the task from the computer system to the programmer/feeder system;

performing the task by the programmer/feeder system independent of the computer system by processing and programming the programmable microdevices;

developing return programmer/feeder information resulting from the programmer/feeder system using the processing information;

returning the return programmer/feeder information to the computer system;

providing processing system setup and shutdown parameters;

providing processing system process-specific parameters;

providing the processing system setup parameters to the legacy processing system and the programmer/feeder system;

providing the processing system shutdown parameters to the programmer/feeder system simultaneously with the processing system parameters;

providing the number of processed programmable microdevices to be output from legacy processing system and the programmer/feeder system;

providing the processing system process-specific parameters to legacy processing system and the programmer/feeder system;

controlling the handling of the programmable microdevices;

programming the programmable microdevices;

providing the processing system shutdown parameters to the legacy processing system;

providing a number of programmable microdevices;

determining the number of programmable microdevices processed;

determining the number of programmable microdevices handled; and

developing statistics from the number of programmable microdevices processed and handled.

14. The method as claimed in claim 13 additionally comprising:

providing a microdevice programming system in the programmer/feeder system, the programmer/feeder system having an on-line connection with said computer system; and

performing the task by the programmer/feeder dependent on the computer system using programming information obtained through the on-line connection.

15. The method as claimed in claim 13 additionally comprising:

providing an operator mode;

using portable memory medium to provide the task in the operator mode to the programmer/feeder system independent from the computer system;

returning return programmer/feeder information in the operator mode using the portable memory medium to the computer system; and

storing the return programmer/feeder information in the computer system.

16. The method as claimed in claim 13 comprising:

providing an administrator mode;

providing the processing and programming information related to the task in the administrator mode;

editing the processing and programming information related to the task in the administrator mode; and

storing the processing and programming information related to the programmable microdevices for the legacy processing system and the programmer/feeder system in the administrator mode.

17. (canceled)

18. (canceled)

19. The method as claimed in claim 13 additionally comprising:

serializing the programmable microdevices; and

maintaining a log of the serialized programmable microdevices.

20. The method as claimed in claim 13 additionally comprising:

combining a plurality of tasks to define a kit; and

performing the programming of a kit in the legacy processing system and the programmer/feeder.

21. The method as claimed in claim 13 additionally comprising:

providing programmable microdevice information;

providing programmer/feeder system setup parameters;

providing format information related to the programmer/feeder system;

inputting the number of processed programmable microdevices to be output from the programmer/feeder system;

providing the programmer/feeder system setup parameters and format to the programmer/feeder system;

transferring the programmable microdevice information from the computer system to the processing system;

transferring the programmer/feeder system form from the computer system to the programmer/feeder system;

processing the programmable microdevices;

obtaining information from the processing of the programmable microdevices; and

transferring the information from the programming of the programmable microdevices.

22. The method as claimed in claim 21 wherein the step of:

transferring includes the use of a portable memory medium.

23. The method as claimed in claim 22 wherein the step of:

transferring includes the use of a local area network connection.

24. The method as claimed in claim 13 including the steps of:

providing an administrator mode; and

protecting provision of the operator mode using a password input in the administrator mode.

25. The method as claimed in claim 13 including the step of:

providing information for affecting changes selected from a group consisting of software, firmware, and a combination thereof by using a portable memory medium.

26. A method for processing microdevices comprising:

providing a computer system having processing information related to the microdevices as a task;

providing a legacy processing system;

providing a non-legacy processing system;

providing the task from the computer system to the legacy processing system with constant interaction therebetween;

providing the task from the computer system to the non-legacy processing system for performing the task by the non-legacy processing system independent of the computer system;

developing return non-legacy information resulting from the non-legacy processing system using the task; and

returning the return non-legacy information to the computer system.

27. The method as claimed in claim 26 additionally comprising:

providing a microdevice programming system in the legacy processing system, the legacy processing system having an on-line connection with said computer system; and

programming the microdevices in the microdevice programming system using the task provided through the on-line connection from the computer system to the processing system.

28. The method as claimed in claim 26 additionally comprising:

providing an operator mode;

providing a microdevice programming system in the non-legacy processing system, the microdevice programming system standing alone from the computer system;

using the processing information for the task in the operator mode in the non-legacy processing system independent from the computer system;

returning return information in the operator mode from the non-legacy processing system using portable medium to the computer system; and

storing the return information in the computer system.

29. The method as claimed in claim 26 additionally comprising:

providing an administrator mode;

providing programming information related to the task in the administrator mode;

editing the processing and programming information related to the task in the administrator mode; and

storing the processing and programming information related to the microdevices for the legacy processing system and the non-legacy processing system as the task in the administrator mode.

30. The method as claimed in claim 26 including additionally comprising:

providing processing system setup and shutdown parameters;

providing processing system process-specific parameters;

providing the processing system setup parameters to the legacy processing system and the non-legacy processing system;

providing the processing system shutdown parameters to the non-legacy processing system simultaneously with the processing system setup parameters;

providing the number of processed microdevices to be output from the legacy processing system and the non-legacy processing system;

providing processing system process-specific parameters to the legacy processing system and the non-legacy processing system;

controlling the handling of the microdevices;

processing the microdevices; and

providing the processing system shutdown parameters to the legacy processing system.

31. The method as claimed in claim 30 additionally comprising:

providing a number of microdevices;

determining the number of microdevices processed;

determining the number of microdevices handled; and

developing statistics from the number of microdevices processed and handled.

32. The method as claimed in claim 30 additionally comprising:

serializing the microdevices; and

maintaining a log of the serialized microdevices.

33. The method as claimed in claim 26 additionally comprising:

combining a plurality of tasks to define a kit; and

performing the processing of a kit through the off-line connection.

34. The method as claimed in claim 26 additionally comprising:

providing microdevice information;

providing processing system setup parameters;

providing format information related to the off-line connection;

inputting the number of processed microdevices to be output from the processing system;

providing the processing system setup parameters and format to the processing system;

transferring the microdevice information from the computer to the processing system;

transferring the processing system format from the computer to the processing system;

processing the microdevices;

obtaining information from the processing of the microdevices; and

transferring the information from the processing of the microdevices.

35. The method as claimed in claim 34 wherein:

transferring includes the use of a portable memory medium.

36. The method as claimed in claim 34 wherein:

transferring includes the use of a direct communication connection.

37. The method as claimed in claim 26 additionally comprising:

providing an administrator mode; and

protecting provision of the operator mode using a password input in the administrator mode.

38. A method for processing and programming programmable microdevices comprising:

providing a computer system having processing information and programming information related to the programmable microdevices combined as a task in the computer system;

providing a legacy processing system;

providing a programmer/feeder system;

providing the task from the computer system to the programmer/feeder system;

performing the task by the programmer/feeder system independent of the computer system by processing and programming the programmable microdevices;

developing return programmer/feeder information resulting from the programmer/feeder system using the processing information; and

returning the return programmer/feeder information to the computer system.

39. The method as claimed in claim 38 additionally comprising:

providing a microdevice programming system in the programmer/feeder system, the programmer/feeder system having an on-line connection with said computer system; and

performing the task by the programmer/feeder dependent on the computer system using programming information obtained through the on-line connection.

40. The method as claimed in claim 38 additionally comprising:

providing an operator mode;

using portable memory medium to provide the task in the operator mode to the programmer/feeder system independent from the computer system;

returning return programmer/feeder information in the operator mode using the portable memory medium to the computer system; and

storing the return programmer/feeder information in the computer system.

41. The method as claimed in claim 38 additionally comprising:

providing an administrator mode;

providing the processing and programming information related to the task in the administrator mode;

editing the processing and programming information related to the task in the administrator mode; and

storing the processing and programming information related to the programmable microdevices for the legacy processing system and the programmer/feeder system in the administrator mode.

42. The method as claimed in claim 38 additionally comprising:

providing processing system setup and shutdown parameters;

providing processing system process-specific parameters;

providing the processing system setup parameters to the legacy processing system and the programmer/feeder system;

providing the processing system shutdown parameters to the programmer/feeder system simultaneously with the processing system parameters;

providing the number of processed programmable microdevices to be output from legacy processing system and the programmer/feeder system;

providing the processing system process-specific parameters to legacy processing system and the programmer/feeder system;

controlling the handling of the programmable microdevices;

programming the programmable microdevices; and

providing the processing system shutdown parameters to the legacy processing system.

43. The method as claimed in claim 42 additionally comprising:

providing a number of programmable microdevices;

determining the number of programmable microdevices processed;

determining the number of programmable microdevices handled; and

developing statistics from the number of programmable microdevices processed and handled.

44. The method as claimed in claim 42 additionally comprising:

serializing the programmable microdevices; and

maintaining a log of the serialized programmable microdevices.

45. The method as claimed in claim 38 additionally comprising:

combining a plurality of tasks to define a kit; and

performing the programming of a kit in the legacy processing system and the programmer/feeder.

46. The method as claimed in claim 38 additionally comprising:

providing programmable microdevice information;

providing programmer/feeder system setup parameters;

providing format information related to the programmer/feeder system;

inputting the number of processed programmable microdevices to be output from the programmer/feeder system;

providing the programmer/feeder system setup parameters and format to the programmer/feeder system;

transferring the programmable microdevice information from the computer system to the processing system;

transferring the programmer/feeder system form from the computer system to the programmer/feeder system;

processing the programmable microdevices;

obtaining information from the processing of the programmable microdevices; and

transferring the information from the programming of the programmable microdevices.

47. The method as claimed in claim 46 wherein:

transferring includes the use of a portable memory medium.

48. The method as claimed in claim 47 wherein:

transferring includes the use of a local area network connection.

49. The method as claimed in claim 38 additionally comprising:

providing an administrator mode; and

protecting provision of the operator mode using a password input in the administrator mode.

50. The method as claimed in claim 38 additionally comprising:

providing information for affecting changes selected from a group consisting of software, firmware, and a combination thereof by using the portable memory medium.